Vol. 4(1), pp. 1-12, 2025
eISSN: 2976-4009

ASEAN Journal of Process Control

Research Article

Enhancing PID Tuning with Routh Stability and Empirical Correlations (RSEC): Performance Comparison and Applications

Jobrun Nandong [1*] and Noraini Mohd[2*]

[1]Department of Chemical and Energy Engineering, Curtin University Malaysia, 98009 Miri, Sarawak, Malaysia

[2]School of Energy and Chemical Engineering, Xiamen University Malaysia, 43900 Sepang, Selangor, Malaysia

*Corresponding Author: jobrun.n@curtin.edu.my , noraini.mohd@xmu.edu.my

Abstract: This paper presents a new rule-based method called the Routh Stability with Empirical Correlation (RSEC) for tuning a PID controller under model or process mismatch. The RSEC method employs a two-stage calibration procedure to accommodate the FOPDT modeling error in the PID tuning. The RSEC calibration utilizes the deadtime to time constant ratio to adjust the dimensionless tuning parameters subject to specified gain and phase margins. It results in empirical correlations relating the dimensionless tuning parameters to the deadtime to time constant ratio. The second calibration establishes correcting factor correlations for the modeling error. Servo and regulatory simulations demonstrated that RSEC produces the smallest ITAE of 0.396 and 0.151 when tested in a cascade process, outperforming the other established control tuning methods, including Skogestad IMC (SIMC). Further application of the developed RSEC method to several examples of SISO and MIMO processes shows the efficacy of the new tuning method compared to the Robust Response Time, AMIGO, and SIMC.

Keywords: PID Control; Process Dynamics; FOPDT Identification; SIMC Tuning; AMIGO Tuning.

1. Introduction

Proportional Integral Derivative (PID) control has become the bread and butter in the process industry since its inception many decades ago. About 80-90% of industrial controllers are PID-type [1], for which about 80% are Proportional Integral (PI) or Proportional (P) while the remaining 20% are full-PID controllers [2]. Despite the emergence of several advanced control techniques, such as the Model Predictive Control (MPC), fixed structure PID-type controllers remain dominating industrial control applications [3]. The reasons for the PID control's widespread acceptance are attributable to several factors: among the most well-known are its easy-to-understand concept, robust performance, and well-established platform for controller implementation. Notwithstanding its acceptance, the continuous challenge for the PID control implementation remains the controller tuning in the face of various process dynamics, disturbance occurrences, and process/model mismatch. Note that the PID controller tuning falls into the non-convex optimization class. Indeed, tuning software has become a big business in the industrial automation sector [4].

Although PID control is simple compared to more advanced techniques, finding good PID controller tuning values is not often straightforward. For that reason, poor PID tuning has made itself a popular culprit for many underperformed process plants. One should note that such poor PID tuning in process plants is not due to the lack of PID tuning formulas and rules available to the designers. On the contrary, researchers have developed numerous PI or PID tuning rules and formulas over decades since the classical Ziegler-Nichols tuning procedure. For example, several well-known PID tuning methods were compared and discussed in [5]. A vast collection of PI or PID controller tuning rules is now available in a handbook [6]. Note that most PID tuning rules require process models where the First-Order plus Deadtime (FOPDT) model is the most common ideal model. For open-loop stable processes, the FOPDT model can adequately explain the behavior of many industrially-proven techniques [7]. However, real processes often deviate markedly from the FOPDT model, which means significant modeling error exists. The modeling error often leads to unpredictable performance of the PI or PID controller designed using the FOPDT model when it is applied in real systems. In [8], the authors highlighted this issue and tried to address the PID ideal tuning rule adjustment to accommodate the process conditions.

The paper aims to introduce a new PID controller tuning technique, which combines the analytical formulas obtained using the Routh stability conditions with a set of empirical equations. The empirical equations are necessary to account for the modeling error in the PID controller tuning based on the ideal FOPDT model. The salient feature of the new technique is to keep the PID controller robustness in terms of gain and phase margins within the desired ranges despite modeling errors. Therefore, the actual PID controller performance implemented in the process will not deviate significantly from the anticipated performance based on the ideal FOPDT model. An advantage of the developed tuning method is that it is deployable in the field environment where the control designer can perform the required calculations manually without a sophisticated computational tool. Also, the tuning rules can accept a crude FOPDT model identified via the visual identification of the process step data. A simple visual identification rule is also suggested in this paper. In the present paper, the new tuning technique is known as the RSEC, abbreviation for Routh Stability with Empirical Correlation.

The rest of the paper is structured as follows. Section 2 presents some preliminaries to provide brief backgrounds to the proposed method. Section 3 outlines the methodology necessary to build the RSEC tuning method and the modeling error assessment approach. Section 4 provides the results and discussion of the developed RSEC method and some examples of its applications to SISO, Cascade, and MIMO processes. Finally, Section 5 highlights the findings and potential future research directions related to the proposed RSEC method.

2. Preliminaries

2.1 SISO feedback control

Figure 1 shows the standard feedback control block diagram of a single-input and single-output (SISO) process where G_c, G_a, G_p and G_s represent the feedback controller, actuator, process, and sensor sub-systems, respectively. Meanwhile, R, E, C, U, D and Y represent setpoint, error, controller output, manipulated variable (e.g., a flow rate), disturbance and controlled variable signals respectively.

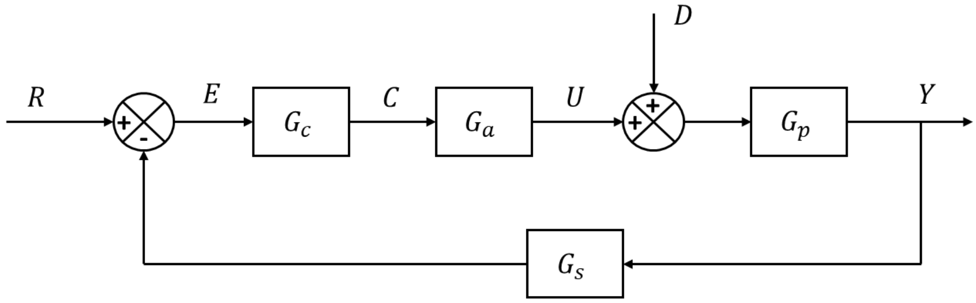

Figure 1. Block diagram of a SISO feedback control.

In Figure 1, the closed-loop setpoint transfer function:

$$H_R(s) = \frac{G_c(s)G_a(s)G_p(s)}{1 + G_c(s)G_a(s)G_p(s)G_s(s)} \tag{1}$$

The characteristics equation from (1) serves as a pathway to develop the proposed RSEC tuning method. It often desirable to develop a model for the lumped $G_aG_pG_s$ as one transfer function, as $G_{pa} = G_aG_pG_s$

for the controller tuning where G_{pa} is obtainable using process data, by fitting it to the First-Order plus Deadtime (FOPDT) model:

$$G_{pa}(s) = \frac{K_p e^{-\theta_p s}}{\tau_p s + 1} \quad (2)$$

In the FOPDT, K_p, τ_p and θ_p denote the process (steady-state) gain, time constant and deadtime, respectively. In many processes, the FOPDT parameters might be sufficient for finding appropriate tuning values via an existing PID tuning rule in the literature.

2.2 Proportional-Integral-Derivative (PID) controller

The ideal PID controller is as follows:

$$G_c(s) = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right) \quad (3)$$

where K_c , τ_I and τ_D denote the controller gain, integral time, and derivative time constants, respectively. To avoid the derivative kick issue, the ideal PID controller is often augmented with a filter (i.e., PIDF controller):

$$G_c(s) = K_c\left(1 + \frac{1}{\tau_I s} + \tau_D s\right)\left(\frac{1}{\tau_f s + 1}\right) \quad (4)$$

where τ_f is the filter time constant.

3. Methodology

Note that the developed empirical tuning correlations come from an extensive calibration and comparison with three well-established PID tuning methods, *viz.* the Skogestad IMC (SIMC) [9]- [10], AMIGO [11] and Robust Response Time (RRT). The MATLAB™ Control System Designer toolbox provides all three tuning methods mentioned.

3.1 PID stability region

The first step is to establish the PID stability region based on the FOPDT model in (2). The details for establishing PID control stability regions using the well-known Routh stability criteria are available in [12]- [13] for unstable and integrating processes. In the present work, the objective is to develop the PID stability region based on the FOPDT model of an open-loop stable (or self-regulating) process. The procedure in the aforementioned papers is applied to obtain the PID stability region of a process that can be approximated by the FOPDT model. To apply the Routh stability conditions in searching for the PID stability region, it is essential to establish a generalized characteristics equation. The generalized characteristics equation J is as follows [12]:

$$J(s) = \underbrace{(h_n + K_L f_n)}_{a_n} s^n + \underbrace{(h_{n-1} + K_L f_{n-1})}_{a_{n-1}} s^{n-1} + \cdots + \underbrace{(h_1 + K_L f_1)}_{a_1} s + \underbrace{(h_0 + K_L f_0)}_{a_0} = 0 \quad (5)$$

In (5), K_L denotes the loop gain, h_j and f_j are coefficients which are functions of model and controller parameters. For comparison, the classical characteristics equation is:

$$J(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_s s + a_0 s^0 = 0 \quad (6)$$

The distinguishing feature of the generalized characteristics equation from the classical one is that the former explicitly shows the loop gain term. Significantly, the loop gain represents the performance of the closed-loop system under the given controller algorithm. Hence, the higher the loop gain, the higher should be the achievable control performance and vice versa. For a PID controller applied to the given process, the loop gain is $K_L = K_c K_p$, assuming the actuator and sensor gains are unity or inclusive in the K_p term. Considering the FOPDT model, the closed-loop characteristics equation for the ideal PID controller:

$$J(s) = \underbrace{0.5\theta_p \tau_I(\tau_p - K_L\tau_D)}_{a_3} s^3 + \underbrace{\tau_I[\tau_p + 0.5\theta_p + K_L(\tau_D - 0.5\theta_p)]}_{a_2} s^2 + \underbrace{[\tau_I + K_L(\tau_I - 0.5\theta_p)]}_{a_1} s + \underbrace{K_L}_{a_0} = 0 \quad (7)$$

where the deadtime term, is approximated using the 1/1 Padě formula: $e^{-\theta_p s} \cong (1 - 0.5\theta_p s)/(1 + 0.5\theta_p s)$. Here, the results of the Routh stability analysis for the ideal PID controller leads to the following stability region:

$$\Omega_{pid} := \begin{cases} K_c = \dfrac{r_p}{K_p}\left(\dfrac{\tau_p}{\tau_D}\right), & r_p \in (0,1) \\ \tau_D = r_d\left[0.5\theta_p + \left(\dfrac{0.5\theta_p\tau_p}{2\tau_p + 0.5\theta_p}\right)\left(\dfrac{1}{r_d} - 1\right)\right], & r_d \in (0,1) \\ \tau_I = r_i\left(\dfrac{\theta_p}{2}\right), & r_i \in (1,\infty) \end{cases} \quad (8)$$

Notice that the stabilizing region above provides a way to tune the PID controller based on the FOPDT model by adjusting the three dimensionless parameters (r_p, r_i, r_d) instead of directly tuning the original PID parameters (K_c, τ_I, τ_D). The dimensionless tuning has an advantage over the conventional direct PID tuning. This is because the dimensionless tuning parameters have values that have clear bounds which is a conveniently practical index. Additionally, the dimensionless parameters can vary according to the deadtime to time constant (DTC) ratio of the FOPDT model, allowing empirical correlations to be established.

3.2 Modeling error assessment

This section outlines the approach to assess the FOPDT modeling error for developing the RSEC tuning method later on. Most industrial processes of interest have more complex dynamics than the FOPDT model can represent, e.g., the process can have several poles and zeros. On the contrary, the FOPDT model only takes one pole without a zero. For that reason, there is often a significant dynamic mismatch between the estimated FOPDT model and the actual process. The mismatch exists even if the model fitness (curve fitting) is very high. The assessment of the model/plant mismatch using the time-domain data is illustrated in Figure 2.

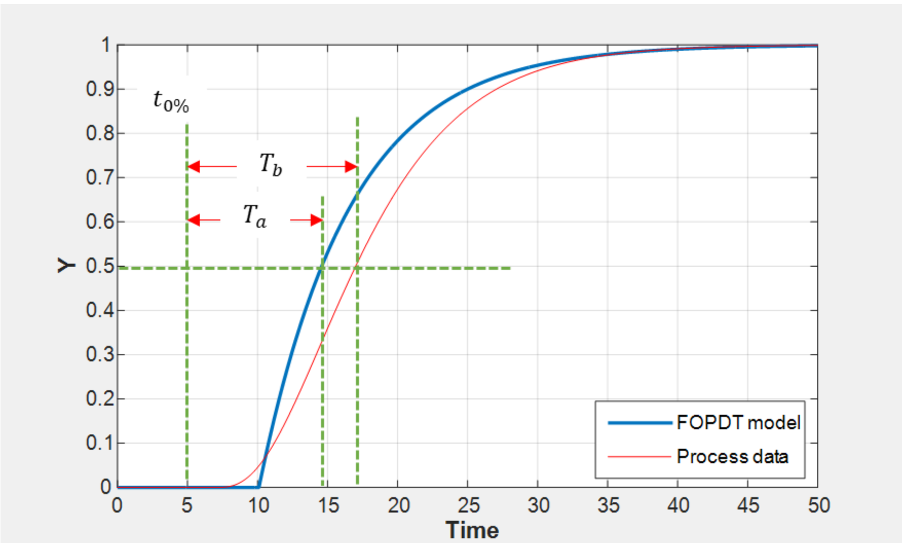

Figure 2. Quantifying the model/process mismatch.

The suggestion is to use the times taken by the identified FOPDT model and step test responses to reach 50% of the complete responses as a threshold point to estimate the modeling error. If the FOPDT model 100% fits the process (step test data), then $T_a = T_b$, otherwise $T_a < T_b$ or $T_a > T_b$.

If $T_a < T_b$, it suggests the overall process dynamics are markedly slower than the corresponding FOPDT dynamics. It happens when the FOPDT model fails to capture the non-dominant modes having faster dynamics (fast modes) relative to the slow dominant process modes: FOPDT time-related parameters mainly capture the dynamics of the process modes with large time constants (slow modes). Note that a process often consists of multiple modes which could be multi-scale in nature [14]. For $T_a > T_b$ it implies that the FOPDT model exhibits slower dynamics than the actual process. This can happen when there are minimum-phase zeros in the actual process which the FOPDT model ignores.

Based on Figure 2, the additional information that might serve as a measure of modeling error is the distance between the model and actual process responses located halfway to the output final steady-state response. The proposed modeling error parameters are defined as follows:

$$Dynamic\ Distance: x_D = T_b - T_a \quad (9)$$

$$Fractional\ Dynamic\ Distance: \sigma_x = \frac{x_D}{T_a} \quad (10)$$

$$Model\ Error\ to\ DTC\ ratio: \rho_{xr} = \frac{\sigma_x}{\varphi_{DT}} \quad (11)$$

where φ_{DT} is the deadtime to time constant (DTC) ratio, calculated as $\varphi_{DT} = \theta_p/\tau_p$. Note that the modeling error parameters in (10) and (11) are dimensionless.

3.3 *Calibration of PID tuning formulas*

The following analytical PID formulas are proposed based on the PID stability region obtained in Section 3.1:

$$\begin{cases} \tau_D = r_d \left[0.5\theta_p + \dfrac{0.5\theta_p \tau_p}{(2\tau_p + 0.5\theta_p)\left(\dfrac{1}{r_d} - 1\right)} \right] \\ K_c = \dfrac{r_p}{K_p}\left(\dfrac{\tau_p}{\tau_D}\right) \\ \tau_I = r_i \tau_p + 0.5\theta_p \end{cases} \quad (12)$$

where the dimensionless tuning parameters take the following bounds: $0 < r_d < 1$, $0 < r_p < 1$ and $r_i > 0$. Notice that, the proposed formulas slightly modify the equation for calculating the integral time constant but it still meets the stability requirement, i.e., $\tau_I > 0.5\theta_p$ that is valid so long that $r_i > 0$. The analytical PID tuning formulas (12) transform the three original PID tuning parameters (K_c, τ_I, τ_D) into three dimensionless tuning parameters (r_p, r_i, r_d). A designer can calibrate the dimensionless parameter values against the DTC ratio. The present paper proposes two-step calibrations, as explained in the following sub-sections.

3.3.1 Calibration based on ideal FOPDT model

The first calibration requires the FOPDT model as follows:

$$G_m(s) = \frac{e^{-\theta_p s}}{10s + 1} \quad where\ \varphi_{DT} = \frac{\theta_p}{10} \in [0.05, 5.0] \quad (13)$$

The first calibration objective is to construct empirical correlations for the dimensionless parameters as functions of the DTC ratio (or φ_{DT}) for the desired ranges of gain margin (GM) and phase margin (PM). The empirical correlation takes the form below:

$$r_d = f_d(\varphi_{DT}) \quad for\ GM \in (8.5dB, 10.5dB), PM \in (53^o, 67^o) \quad (14)$$

The target ranges of GM and PM are 8.5-10.5dB and 53-67^o respectively.
The following general steps apply to the calibration procedure:

Step 1: Fix the DTC ratio and select values for the three dimensionless tuning parameters so that the PID controller attains the desired GM and PM ranges.
Step 2: While keeping the r_p and r_i values in Step 1, refine the r_d value so that the setpoint tracking overshoot is between 1% and 5% (desired OS is 3%).
Step 3: Design three other PID controllers via the RRT, AMIGO, and SIMC methods.
Step 4: Compare the proposed PID controller's performance with those obtained via the RRT, AMIGO, and SIMC methods.
Step 5: If the proposed PID controller outperforms all the other three PID controllers, accept the r_d value for the given DTC ratio. Else, go back to Step 2 or Step 1 to re-adjust the r_p, r_i, and r_d values, then repeat Steps 4-5.

The performance assessment is based on the Time Integral Absolute Error (ITAE) criterion. The DTC ratio for the lag-dominant regime is $\varphi_{DT} < 0.9$ and the deadtime-dominant is $\varphi_{DT} \geq 0.9$.

3.3.2 Modeling error correction factors

The process model structure for the second calibration:

$$G_p(s) = \frac{e^{-\theta_p s}(\tau_{z1}s + 1)^{m_1}(\tau_{z2}s + 1)^{m_2}}{(\tau_1 s + 1)^{n_1}(\tau_2 s + 1)^{n_2}(\tau_3 s + 1)^{n_3}} \tag{15}$$

where $\tau_1 > \tau_2 > \tau_3$, $\tau_z \geq 0$, $\theta_p \geq 0$, n_j and m_i are integers. Assume the FOPDT model can approximate the process above, i.e., using the visual identification or other techniques available. The goal of the second calibration is to develop correcting factor correlations as functions of x_D, σ_x and ρ_{xr}. The general steps in Section 3.3.1 is adapted in this second calibration.

4. Results and discussion

4.1 Tuning correlations

For the first regime, two variants of the RSEC method are developed: RSEC1-vs1 and RSEC1-vs2, which correspond to without and with correction factor correlations, respectively. Only the RSEC2-vs1 variant is developed for the deadtime-dominant processes.

4.1.1 Case 1 (lag dominant): $\boldsymbol{\varphi_{DT} = \theta_p/\tau_p < 0.9}$

Case 1.1: without correction factor (RSEC1-vs1)
The following tuning rules are established for the PID analytical formulas in (12):

$$\begin{cases} r_p = 0.25 \\ r_i = 0.8 \\ r_d = b_3\varphi_{DT}^3 + b_2\varphi_{DT}^2 + b_1\varphi_{DT} + b_0 \end{cases} \tag{16}$$

where $b_0 = 0.8441$, $b_1 = -0.8928$, $b_2 = 0.6737$ and $b_3 = -0.2468$. The tuning rules in (16) aim to attain GM and PM in the ranges of 8.5-10.5dB and 53-67° with an overshoot (OS) below 5%. The given PID or PIDF controller based (16) can outperform those designed via the RRT, AMIGO and SIMC when tested for various DTC ratios.

Case 1.2: with correction factors (RSEC1-vs2)
To account for the FOPDT modeling error (non-ideal process condition), at least one empirical correction factor correlation is necessary for modifying one of the dimensional tuning parameters (r_p, r_d or r_i value) of the ideal tuning rules in (16). In this work, two empirical correction factors are developed for the two dimensionless tuning parameters: r_p and r_d but no correction factor is required for r_i, Here, r_i is fixed at 0.8 as in Case 1.1. The corrected dimensionless parameters (r_p' and r_d') are expressed as follows:

$$\begin{cases} r_p' = C_{fp} r_p \\ r_d' = C_{fd} r_d \end{cases} \tag{17}$$

where the correction factor for r_p' is,

$$C_{fp} = \begin{cases} 1 + \rho_{xr} \quad for\ 0 < \sigma_x < 0.45 \\ 1 \quad for\ \sigma_x < 0 \end{cases} \tag{18}$$

and for r_d':

$$C_{fd} = \begin{cases} 1 \quad for\ \sigma_x < 0.15 \\ 1 + (b_4\sigma_x^4 + b_3\sigma_x^3 + b_2\sigma_x^2 + b_1\sigma_x + b_0)\rho_{xr} \end{cases} \tag{19}$$

In (19), $b_0 = -3.3772$, $b_1 = 53.659$, $b_2 = -319.23$, $b_3 = 926.19$ and $b_4 = -907.95$. This correlation must be used within the given range, $0.15 \le \sigma_x \le 0.45$. Additionally, it is also advisable to correct the ideal FOPDT time constant τ_p as follows:

$$\tau_p' = \begin{cases} \tau_p + 2x_D \quad for\ 0 \le \sigma_x \le 0.45 \\ \tau_p + x_D \quad for\ \sigma_x < 0 \end{cases} \tag{20}$$

where x_D denotes the dynamic distance between the model and the given process as in (9).

For the PID controller with filter in (4) or PIDF, the filter time constant τ_f is taken to be 10% of the derivative time τ_D (i.e., $\tau_f = 0.1\tau_D$). This simple rule is recommended for all cases without or with correction factors in both lag-dominant and deadtime dominant regimes.

For reducing a potentially large OS in the setpoint tracking, it is desirable to implement a setpoint pre-filter designed as follows:

$$F_{sp}(s) = \begin{cases} \dfrac{1}{\left(\frac{\tau_D}{\varphi_{DT}}\right)s + 1} \quad for\ \dfrac{\tau_I}{\tau_D} \ge 3 \\ \dfrac{\varphi_{DT}s + 1}{\tau_D s + 1} \quad for\ \dfrac{\tau_I}{\tau_D} < 3 \end{cases} \tag{21}$$

The PIDF controller designed using the RSEC1-vs2 can outperform the PID controllers tuned using the RRT, AMIGO and SIMC methods in various case studies.

4.1.2 Case 2 (deadtime-dominant): $0.9 \le \varphi_{DT} < 5.0$

Case 2.1: without correction factors (RSEC2-vs1)

In the deadtime-dominant regime, the following correlations (RSEC2-vs1) are developed:

$$\begin{cases} r_p = 0.25 \\ r_i = -0.1756\varphi_{DT} + 0.8803 \\ r_d = b_4 + \varphi_{DT}^4 + b_3\varphi_{DT}^3 + b_2\varphi_{DT}^2 + b_1\varphi_{DT} + b_0 \end{cases} \tag{22}$$

where $b_0 = 0.7518$, $b_1 = -0.4859$, $b_2 = 0.1722$, $b_3 = -0.0303$ and $b_4 = 0.0021$.

The setpoint pre-filter as in (21) is still applicable in this regime to reduce the OS value following the setpoint change. For σ_x between -0.1 and 0.1, the RSEC2-vs1 can be applied without incurring a significant loss of the expected performance compared to the ideal FOPDT case. Also, it is noticed that in this regime, σ_x is relatively small and insensitive to the fast dynamics often ignored by the approximated FOPDT model.

4.2 Illustrative examples

4.2.1 SISO example 1 – lag-dominant process

Consider a process given as follows:

$$G_p(s) = \frac{1.6(0.7s+1)e^{-2.5s}}{(7.5s+1)(3s+1)(s+1)} \quad (23)$$

There are two FOPDT models for the process in (25): one based on visual inspection and another via MATLAB System Identification. The controller designed via the RSEC1-vs2 adopts the FOPDT based on the visual identification, but controllers designed based on the RRT, AMIGO, and SIMC use the one based on the System Identification. The servo and regulatory control performances (Table 1) indicate that the PIDF controller designed via RSEC1-vs2 can outperform the controllers designed via the RRT, AMIGO, and SIMC methods.

Table 1. The PIDF performances ($ITAE \times 10^{-1}$) for the SISO example 1.

Method	Servo	Regulatory
RRT	18.27	107.0
AMIGO	33.55	124.2
SIMC	11.2	110.9
RSEC1-vs2$	6.845	101.2
RSEC1-vs1a$$	10.15	83.89
RSEC1-vs1b$$#	11.49	136.5

$using G_{a2} $$using G_{a1} $$#using G_{a2}

The performance test is based on a 1-unit step change in the setpoint or input disturbance. Another observation is that the PIDF controller tuned using the RSEC1-vs1 and the visual inspection's FOPDT model exhibits a lower performance than the controller tuned using the RSEC1-vs2. It is due to the significant modeling error associated with the visual inspection's FOPDT. Hence, the ideal tuning rule (RSEC1-vs1) gives an overly conservative performance – GM is too high. However, the RSEC1-vs1 applies to the System Identification's FOPDT model yields a PIDF controller that provides a comparable performance with the controller obtained via the RSEC1-vs2. Figure 3 shows the comparative closed-loop responses of all the PIDF controllers. To reduce the OS, one can apply the setpoint pre-filter designed via the formula in (21):

$$F_{sp}(s) = \frac{1}{3s+1} \; where \; \frac{\tau_I}{\tau_D} = 6.1, \varphi_{DT} = 0.599 \quad (24)$$

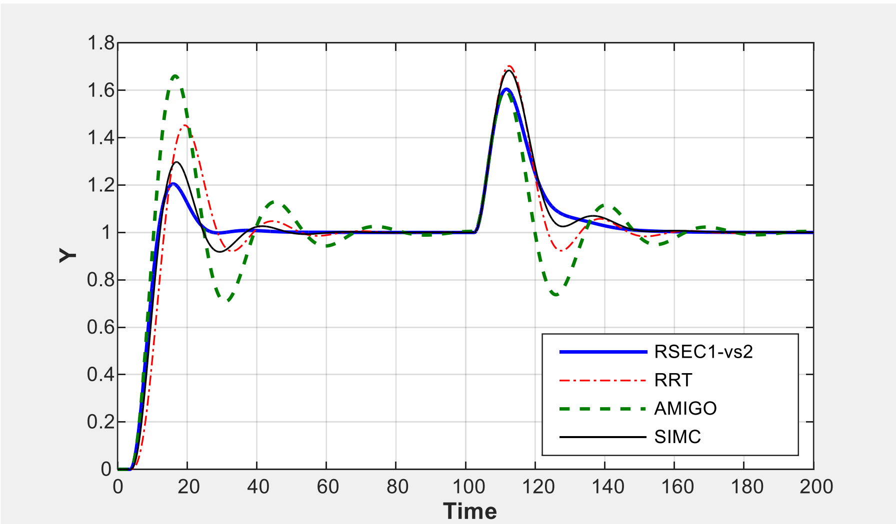

Figure 3. Setpoint tracking and regulatory responses for SISO example 1.

The setpoint pre-filter reduces the OS from about 20% to 5%, well within the recommended range of 13% or below as suggested in [11].

4.2.2 SISO example 2 – deadtime-dominant process

Consider a deadtime-dominant process:

$$G_p(s) = \frac{e^{-10s}}{(5s+1)^2(2s+1)} \tag{25}$$

Since the process is a deadtime-dominant, the RSEC2-vs1 is the preferred choice to tune the PIDF controller. Table 2 displays the ITAE values of the PIDF controllers obtained using the RSEC2-v1, AMIGO, SIMC, and RRT. The performance test is subject to a 1-unit step change in the setpoint or input disturbance. The proposed PIDF controller (RSEC2-vs1) outperforms all other controllers tuned using RRT, AMIGO, and SIMC methods. Figure 4 shows the comparative closed-loop responses for the different PIDF controllers subject to a 1-unit step changes in the setpoint and disturbance.

Table 2. PIDF performances ($ITAE \times 10^{-1}$) for the SISO example 2.

Method	Servo	Regulatory
RRT	79.52	106.9
AMIGO	48.16	96.54
SIMC	48.34	115.1
RSEC2-vs1	41.24	73.20

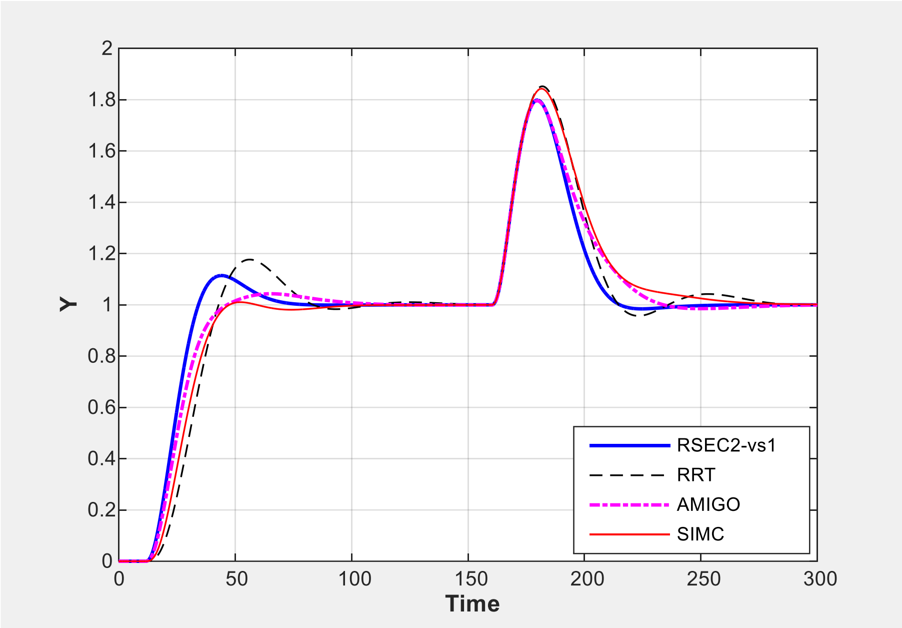

Figure 4. Servo and regulatory control responses for the SISO example 2.

The PIDF controller obtained using the RSEC2-vs1 exhibits about 11% OS, hence it is within the acceptable range. Still, one can reduce the OS by applying a setpoint pre-filter designed via the rule in (21):

$$F_{sp}(s) = \frac{1}{2.586s+1} \quad where \ \frac{\tau_I}{\tau_D} = 3.25, \qquad \varphi_{DT} = 1.45 \tag{26}$$

The pre-filter reduces the OS from about 11% down to 3%. Another benefit of using the setpoint pre-filter is the reduction in the peak control output, which can help reduce wear-and-tear in the actuator system.

4.2.3 Cascade processes

Consider the primary and secondary processes given by:

$$\begin{cases} G_{p1}(s) = \dfrac{e^{-s}}{(16s+1)(7s+1)(3s+1)^2} \\ G_{p2}(s) = \dfrac{2e^{-2s}}{(4s+1)(s+1)} \end{cases} \tag{27}$$

Table 3 displays the normalized ITAE values of all cascade control systems for the servo and regulatory control objectives. The cascade control system based on the RSEC1-vs2 method exhibits the best performance. Figure 5 displays the closed-loop responses of primary output corresponding to the cascade control systems. The cascade control system based on the RSEC1-vs2 gives about 13% overshoot. However, the peak control output is substantially large for the control system involved.

Table 3. PID controllers and performances ($ITAE \times 10^{-1}$) for the cascade process.

Method	**Servo**	**Regulatory**
RRT	8.902	2.619
AMIGO	7.717	2.262
SIMC	6.623	3.178
RSEC1-vs2	3.963	1.505

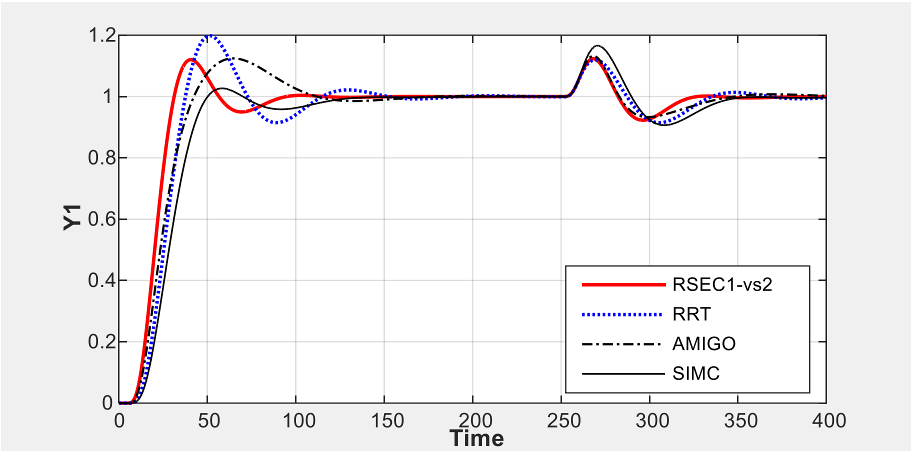

Figure 5. Setpoint tracking and regulatory responses of the cascaded PIDF controllers.

4.2.4 MIMO process

For a demonstration on the effectiveness of the proposed RSEC tuning technique, consider the well-known 2x2 Wood and Berry (WB) distillation process [15]:

$$G(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \tag{28}$$

For simplicity, it is suggested to adopt the Relative Gain Array (RGA) analysis to determine suitable controller pairings. As a result, the RGA matrix is obtained:

$$\Lambda = \begin{bmatrix} 2.0094 & -1.0094 \\ -1.0094 & 2.0094 \end{bmatrix} \tag{29}$$

Based on (29), the RGA matrix recommends direct controller pairings (i.e., $u_1 \sim y_1 / u_2 \sim y_2$) to minimize the impacts of process interactions on control performance. This means that the designer should use input 1 (u_1) to control output 1 (y_1) and so on. The independent multi-loop tuning approach in [16] is the preferred choice for this example. According to the independent tuning approach, it is necessary to establish Effective Open-Loop Transfer Functions (EOTFs) based on the transfer functions in (28). Then, use the EOTFs to design desired controllers independently. However, the EOTF is often a high-order system with multiple deadtime terms. Several methods are available to reduce the EOTF to a simple low-order model. The present work proposes to use the same visual inspection rule for the SISO case to approximate the EOTF to a FOPDT model. Alternatively, use MATLAB System Identification to obtain simpler models; in this case, third-order models that give above 90% fits.

The proposed RSEC1-vs2 is preferable, as it can adjust the ideal tuning to the modeling error. Table 4 shows the multiloop PIDF controller performances (IAE values) corresponding to the RSEC1-vs2, AMIGO, SIMC, and RRT methods. Again, the RSEC1-vs2 gives the best control performance for the 2x2 WB process. Meanwhile, the SIMC's multi-loop PIDF controllers give acceptable performance based on the IAE values.

Table 4. Multi-loop PIDF controller performances (IAE) for different methods.

Method	Y_1	Y_2
SIMC	5.350	20.42
RRT	22.01	50.19
AMIGO	11.04	24.63
RSEC1-vs2	5.821	16.32

However, the first-loop response is too oscillatory. Other methods (AMIGO and RRT) unable to provide a practical performance - too sluggish. Note that the control performance depends on the model used for tuning. It is possible to attain better controller performances using the RRT and AMIGO based on different approximated models. However, this matter is beyond the current scope of research.

5. Conclusion

The new RSEC1-vs2 tuning rules for PID or PIDF tuning apply to SISO, cascade, and multivariable processes. The rules use a set of modeling error parameters which are easy to identified based on the process step data. Also, the ideal RSEC1-vs or RSEC2-vs1 is effective if the process/model mismatch is low. The RSEC method is simple because it only requires a rough FOPDT model estimated using the given step test data. There is no need for a sophisticated tool since the tuning rules only require simple calculations based on the given empirical correlations. Possible future research directions are to develop other variants to cope with integrating and unstable processes. Also, the existing method might be adaptable to online PID tuning applications allowing for correction in the process/model mismatch.

Acknowledgement

This work was partly supported by the Intelligent Systems, Design and Control research cluster, Curtin University Malaysia.

References

[1] Aoyama J, Konishi K, Yamamoto T, Hinamoto T. A BMI based design for robust PID controllers with two-degrees-of-freedom structure. Proceedings of the 2004 American Control Conference, vol.2; 2004. p. 4879-4884.

[2] Luyben ML, Luyben WL. Essentials of process control. New York: McGraw-Hill; 1997.

[3] Åström KJ, Hagglund T. The future of PID control. Control Eng. Pract. 2001; 9(11): 1163-1175. https://doi.org/10.1016/S0967-0661(01)00062-4.

[4] Somefun OA, Akingbade K, Dahunsi F. The dilemma of PID tuning. Annu. Rev. Control 2021; 52: 65-74. https://doi.org/10.1016/j.arcontrol.2021.05.002.

[5] Foley MW, Julien RH, Copeland BR. A comparison of PID controller tuning methods. Can. J. Chem. Eng. 2005; 83(4): 712-722. https://doi.org/10.1002/cjce.5450830412.

[6] O'dwyer A, Handbook of PI and PID controller tuning rules. World Scientific; 2009.

[7] Seborg DE, Edgar TF, Mellichamp DA, Doyle III FJ. Process dynamics and control. John Wiley & Sons; 2016.

[8] Leva A, Maggio M. A systematic way to extend ideal PID tuning rules to the real structure. J. Process Contr. 2011; 21(1): 130-138. https://doi.org/10.1016/j.jprocont.2010.10.014.

[9] Skogestad S. Probably the best simple PID tuning rules in the world. AIChE Annual Meeting, Reno, Nevada, vol. 77; 2001.

[10] Skogestad S, Grimholt C. The SIMC method for smooth PID controller tuning. In: PID Control in the Third Millennium: Lessons Learned and New Approaches, Springer; 2012. p. 147-175.

[11] Åström KJ, Hagglund T. Revisiting the Ziegler-Nichols step response method for PID control. J Process Contr. 2004;14(6):635-650. https://doi.org/10.1016/j.jprocont.2004.01.002.

[12] Seer QH, Nandong J. Stabilization and PID tuning algorithms for second-order unstable processes with time-delays. ISA T. 2017;67:233-245. https://doi.org/10.1016/j.isatra.2017.01.017.

[13] Seer QH, Nandong J. Stabilising PID tuning for a class of fourth-order integrating nonminimum-phase systems. Int. J. Control 2019;92(6):1226-1242. https://doi.org/10.1080/00207179.2017.1387289.

[14] Nandong J, Zang Z. High-performance multi-scale control scheme for stable, integrating and unstable time-delay processes. J. Process Contr. 2013; 23(10): 1333-1343. https://doi.org/10.1016/j.jprocont.2013.08.007.

[15] Luyben WL. Simple method for tuning SISO controllers in multivariable systems. Ind. Eng. Chem. Proc. DD. 1986; 25(3): 654-660, 1986. https://doi.org/10.1021/i200034a010.

[16] Vu TNL, Lee M. Independent design of multi-loop PI/PID controllers for interacting multivariable processes. J. Process Contr. 2010; 20(8): 922-933. https://doi.org/10.1016/j.jprocont.2010.06.012.